What effects topological changes in dynamic graphs?

Mehdi Kaytoue, Yoann Pitarch, Marc Plantevit, Céline Robardet

Noname manuscript No.
(will be inserted by the editor)

What effects topological changes in dynamic graphs?

Elucidating relationships between vertex attributes and the graph structure

Mehdi Kaytoue · Yoann Pitarch · Marc Plantevit · Céline Robardet

Received: date / Accepted: date

Abstract To describe the dynamics taking place in networks that structurally change over time, we propose an approach to search for vertex attributes whose value changes impact the topology of the graph. In several applications, it appears that the variations of a group of attributes are often followed by some structural changes in the graph that one may assume they generate. We formalize the triggering pattern discovery problem as a method jointly rooted in sequence mining and graph analysis. We apply our approach on three real-world dynamic graphs of different natures – a co-authoring network, an airline network, and a social bookmarking system – assessing the relevancy of the triggering pattern mining approach.

Keywords Data mining · Mining methods and analysis · Attributed graph mining · Topological patterns · Dynamic graphs.

1 Introduction

In the last years, graph mining has become a critical area of research but also an important tool for uncovering phenomena hidden in social networks. It allows a better understanding of their nature but also of the Human interactions and behaviors

This work has been partially supported by the project GRAISearch – EU Marie Curie Actions – FP7-PEOPLE-2013-IAPP.

M. Kaytoue
INSA-Lyon, CNRS, LIRIS UMR5205, F-69621, France, E-mail: mehdi.kaytoue@insa-lyon.fr

Y. Pitarch
Université de Toulouse, CNRS, IRIT UMR5505, F-31071, France, E-mail: Yoann.Pitarch@irit.fr

M. Plantevit
Université Claude Bernard Lyon 1, CNRS, LIRIS UMR5205, F-69621, France,
E-mail: marc.plantevit@liris.cnrs.fr

C. Robardet
INSA-Lyon, CNRS, LIRIS UMR5205, F-69621, France, E-mail: celine.robardet@insa-lyon.fr

on the Web, and provides a support for many tasks such as social recommendations [15], community discovery [11], social influence propagation [12], and link prediction [4]. Indeed, real-world phenomena such as social interactions, are often depicted by graphs whose vertices represent entities and edges represent their relationships or interactions. With the rapid development of social media, sensor technologies and bioinformatic assay tools, such kind of graph abstraction has become ubiquitous. By nature, most of these systems are dynamic. Vertices and edges may appear or disappear in time. Besides, the status of a vertex is often described by attributes whose values also change over time. A timely challenge is thus the design of effective *graph mining* methods to discover actionable insights in such *dynamic attributed graphs*, to bring new knowledge on the common rules that govern the networks transformations.

In data mining, dynamic graphs have been analyzed from two main research tracks: (a) the study of the properties that describe the topology of the graph [7,31], or (b) the extraction of specific sub-graphs to describe the graph evolution [2,28,34]. Very few approaches [8] extract patterns that combine information about attribute values and graph topology, but fail to identify the temporal relationships that may exist between the changes of these two components. In this paper we strive to elucidate the temporal relationships between the evolution of vertex attribute values and the graph structure. Let us first illustrate this idea. Figure 1 depicts a social network whose users are linked if they mutually follow their blogs and evolves over 6 timestamps. Attributes a, b, and c denote the number of status updates, positive opinions sent to others and negative opinions received from other users. At each timestamp, each vertex is also naturally provided with topological properties giving his role in the current graph (centrality measures, clustering coefficient, etc.). To ease reading, only vertex degrees deg are represented. From this dynamic attributed graph, we consider that an attribute (or a topological property) strongly varies for a given vertex if the absolute difference between its current value and the one at the previous timestamp is at least of two. These variations are represented by dotted lines. It makes it possible to represent temporal relationships between the vertex attributes and their topological properties as a *triggering pattern*: a sequence of variations that is supported by a given proportion of vertices. In the example, such a sequence is denoted as $\langle\{a^+, b^+\}, \{c^-\}, \{deg^+\}\rangle$ and is supported by the two vertices u_1 and u_3. It can be interpreted as *updating his status more often while giving positive opinions about others and then receiving less negative opinions from the others is often followed by an increase of user's popularity.*

We define the problem of finding temporal relationships between the vertex attributes and their topological properties as the *triggering pattern mining problem*. A given (possibly oriented) dynamic attributed graph is mapped into sequences of variations of both vertex attributes and topological properties. Each sequence is thus related to a vertex and describes its history. We consider this information to identify the sequences of attribute variations that are generally followed by a topological variation.

To this end, we use the notion of emerging patterns in the framework of supervised descriptive rule discovery [9,24] where class labels are given by topological variations. To assist an in-depth analysis, we propose to examine the vertices that supports such an emerging subsequence, by considering several aspects:

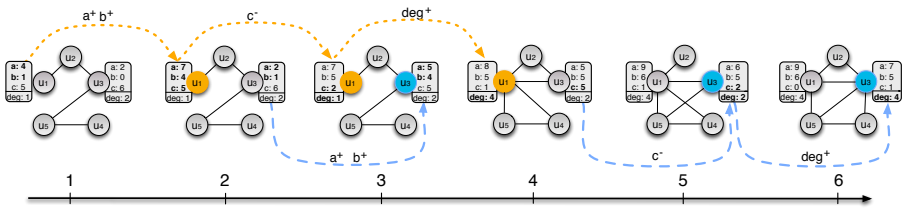


Fig. 1: A dynamic attributed graph on 6 timestamps entailing the triggering pattern $\langle\{a^+, b^+\}, \{c^-\}, \{deg^+\}\rangle$. To ease reading, only attributes for vertices u_1 and u_3 are shown. Colors and bold attribute values indicate strong variations for supporting vertices.

How many vertices support the subsequence? Are those vertices highly connected? Do they convey a high diffusion potential? Are the variations synchronized or spread through time as in Figure 1? Therefore, during the extraction, the mining algorithm has to consider not only the supporting vertices of the subsequence (e.g. [32]), but also the structure of the subgraph induced by them. Additionally, we enhance patterns with semantics, through the definition of several measures and constraints. Furthermore, we introduce a particular representation of sequential patterns (*prefix-closed patterns*), that are the best non-redundant and most discriminant patterns highlighting temporal relationships.

The main contributions of this paper are threefold:

- The introduction of a novel problem, the discovery of triggering patterns of topology changes, defined as a suitable mathematical notion for the study of dynamic networks;
- The design of `TRIGAT`, an efficient algorithm jointly rooted in sequence mining and graph analysis that benefits from various properties of closed sequential pattern mining;
- An extensive empirical study that evaluates the efficiency and the effectiveness of the devised algorithm on three real-world dynamic graphs of different natures: A co-authoring network, an airline network, and a social bookmarking system.

The rest of this paper is organized as follows. The related work is reviewed in Section 2. Section 3 states the pattern domain defined to capture dynamic graph topology changes and their potential causes. Section 4 introduces several measures and constraints used to enrich the semantics of the extracted patterns. Section 5 presents an efficient algorithm designed to mine triggering patterns and Section 6 reports experimental results. The last section concludes the paper.

2 Related Work

Many works dealing with dynamic graphs have been proposed for a decade aiming at characterizing either the graph evolution by focusing on some topological properties [31], or the graph evolution by means of patterns/rules. Borgwardt et al. [3]

introduce the problem of mining frequent sub-graphs in dynamic graphs, i.e. isomorphic graphs that appears in consecutive timestamps. In [17], Lahiri and Berger-Wolf also extract frequent sub-graphs but at periodic or near-periodic timestamps. Inokuchi and Washio [14] define frequent induced subgraph subsequence, i.e. subgraph subsequence whose isomorphic occurrences appear frequently in a graph sequence collection. In [26], the authors extract frequent planar sub-graphs in a sequence of planar graphs and propose spatio-temporal patterns extracted from the occurrence graph. [28] proposes an algorithm to extract evolving patterns, i.e. pseudo-cliques which appear in consecutive timestamps with slight evolutions. [1] mines the evolution of conserved relational states, i.e. sequences of time-conserved pattern on consecutive time. Yang et al. devise an algorithm to identify the most frequently changing component [33]. You and Cook [34] compute graph rewriting rules that describe the evolution between consecutive graphs. These rules are then abstracted into patterns representing the dynamics of graphs. Berlingerio et al. [2] extract patterns based on frequency and derive evolution rules to solve prediction problems [4]. In [23], the authors study dynamic graphs by means of descriptive n-ary association rules.

All the above works only focus on the graph structure and its evolution. They do not take into account the existence of attributes related to the vertices. Several approaches have been proposed to discover new insights in static attributed graphs with several applications in social or biological networks but also in computer vision [36]. The pioneering work [20] proposes a method to find dense homogeneous subgraphs (i.e., subgraphs whose vertices share a large set of attributes). Similar to this work, Günnemann et al. [13] propose a method based on subspace clustering and dense subgraph mining to extract non redundant subgraphs that are homogeneous with respect to vertex attributes. In [37], the authors propose a new method to partition a graph based on both structural and attribute similarities through a unified distance measure. Silva et al. [30] extract pairs of dense subgraphs and Boolean attribute sets such that the Boolean attributes are strongly associated with the dense subgraphs. Similarly in [21], the authors introduce the problem of mining maximal homogeneous clique sets. Another approach is presented in [16] in which the authors propose a probabilistic approach to both construct the neighborhood of a vertex and propagate information into this neighborhood. Following the same motivation, Sese et al. [29] extract (not necessarily dense) subgraph with common itemsets. In [27], the authors propose to mine the graph topology of a large attributed graph by finding regularities among vertex descriptors that are of two types: vertex attributes and topological properties.

None of work mentioned above attempts to rely the graph structure change to the one of vertex attribute values. In [8], the authors define a new kind of pattern that relies on the graph structure and the temporal evolution of the attribute values. It enables to discover set of vertices satisfying a maximum diameter constraint that follow the same trends w.r.t. some attributes. However, these patterns lack expressiveness. They cannot characterize local structure changes by sequence of trends. Indeed, these patterns cannot depict attribute that follow several variations through time.

3 Mining topological changes

A dynamic attributed graph is a sequence of attributed graphs where each vertex takes a value for all the different attributes. Values vary in time as well as edges that may appear or disappear.

Definition 1 (Dynamic attributed graph) Let $\mathcal{G} = \{G_1, \ldots, G_t\}$ be a sequence of t static attributed graphs $G_i = (V, E_i, F)$ with $T = \{1, \ldots, t\}$ the set of timestamps, V the set of vertices, E_i the set of edges that connect vertices of V at time $i \in T$ ($E_i \subseteq V \times V$) and F the set of numerical attributes that map each vertex-time pair to a real value: $\forall f \in F,\ f : V \times T \to \mathbb{R}$.

The structure of each static graph can be characterized by some topological measures [10,19,27] that convey important information on the connectivity of the vertices within the graph at different granularity levels. For instance, the degree describes the close neighborhood of a vertex, whereas the centrality measures depict the role of the vertex in the whole graph. We denote by M the set of topological measures m, with $m : V \times T \to \mathbb{R}$.

Example 1 Figure 1 illustrates a dynamic attributed graph $\mathcal{G} = \{G_1, ..., G_6\}$, with $t = 6$ timestamps. Each graph $G_i = (V, E_i, F)$ shares the set of vertices $V = \{u_1, ..., u_5\}$ and the set of attributes $F = \{a, b, c\}$. The fact that the vertex u_3 takes the value 6 for the attribute c at timestamp 2 is written $c(u_3, 2) = 6$. The topological measure used in this example is the vertex degree: $M = \{deg\}$ and $deg(u_3, 6) = 4$.

A change in the structure of the dynamic graph is observed through a strong variation on some topological measures. Our hypothesis is that those changes can be the consequences of strong variations on vertex attribute values. Let $D = F \cup M$ be the set of vertex descriptors that are either attributes or topological measures. Our goal is to highlight how variations of descriptor values of a vertex can later impact on its connectivity, that is to say variations on descriptions of D followed by variations on measures of M.

To characterize the vertex descriptor variations, an appropriate discretization has to be used. As such, we can represent, for each vertex, its behavior with a sequence of descriptor values variations between any two consecutive time stamps. This procedure has to be wisely chosen with respect to the goals and the properties of the dynamic graph. This choice has no impact on our problem definition: to ease reading of this section we consider the naive discretization of Example 2 that turns any attribute value into an element of $S = \{+, -, \emptyset\}$ to denote increase, decrease or no variation.

Definition 2 (Discretization function) Let $discr : V \times D \times T \to S$ be a discretization function with S a set of variation symbols. In the following, we call an element (d, s) from $D \times S$ a descriptor variation, denoted d^s.

Example 2 Let $S = \{+, -, \emptyset\}$ be a simple discretization function defined as, with $v \in V$, $d \in D$, i an index of T:

$$discr(v, d, i) = \begin{cases} + \text{ if } d(v, i) - d(v, i-1) \geq 2 \quad \text{and } i > 1 \\ - \text{ if } d(v, i) - d(v, i-1) \leq -2 \text{ and } i > 1 \\ \emptyset \text{ otherwise} \end{cases}$$

The dynamic graph can thus be viewed as a set of vertex descriptive sequences defined as follows:

Definition 3 (Vertex descriptive sequence) The set of all variations for a vertex v at time i is a set $vars(v,i) = \{d^{discr(v,d,i)}, \forall d \in D\}$. A vertex v is described by a sequence $\delta(v) = \langle vars(v,2), \ldots, vars(v,t)\rangle$. We note $\Delta = \{\delta(v) \mid v \in V\}$ the set of all sequences.

Example 3 The graph in Figure 1 is transformed into

$$\begin{aligned}\Delta &= \{\\ \delta(u_1) &= \langle\{a^+, b^+, c^\emptyset, deg^\emptyset\}, \{a^\emptyset, b^\emptyset, c^-, deg^\emptyset\}, \{a^\emptyset, b^\emptyset, c^\emptyset, deg^+\}, \{a^\emptyset, b^\emptyset, c^\emptyset, deg^\emptyset\}, \{a^\emptyset, b^\emptyset, c^\emptyset, deg^\emptyset\}\rangle,\\ \delta(u_3) &= \langle\{a^\emptyset, b^\emptyset, c^\emptyset, deg^\emptyset\}, \{a^+, b^+, c^\emptyset, deg^\emptyset\}, \{a^\emptyset, b^\emptyset, c^\emptyset, deg^\emptyset\}, \{a^\emptyset, b^\emptyset, c^-, deg^\emptyset\}, \{a^\emptyset, b^\emptyset, c^\emptyset, deg^+\}\rangle\\ &\}\end{aligned}$$

For the sake of simplicity, we drop any description variation with the symbol $(.)^\emptyset$, i.e. we exclude the description of attribute that has no variation between two consecutive time stamps. We obtain for the vertex u_1 the descriptive sequence $\delta(u_1) = \langle\{a^+, b^+\}, \{c^-\}, \{deg^+\}\rangle$. It means that we observe, for this vertex, an increase of the values of attribute a and b simultaneously, then an decrease of the value of c and finally an increase of the degree of the vertex (deg). Note that in this sequential representation, we do not encode the time stamps of the variations, but simply keep their ordering of appearance. This allows to find later patterns that generalize both (almost) synchronous and asynchronous vertex behaviors. The notion of synchronicity is formalized in the next section.

Now that we properly encode the vertices temporal behavior with a sequence database, we are looking for possible explanations of a topological variation. Such an explanation is given by a sub-sequence (or sequential pattern [32]) of descriptors variations L that happen before the considered topological variation R. It follows that a *triggering pattern* is formalized as a sequence $\langle L, R\rangle$ which is supported by a certain number of vertices. It makes it possible to represent temporal dependencies between D and M, and the confidence or strength of such dependency has to be high.

Definition 4 (Triggering pattern) A triggering pattern is a sequence $P = \langle L, R\rangle$ where L is a sequence of sets of descriptor variations $L = \langle X_1, \ldots, X_k\rangle$ with $\forall j \leq k$, $X_j \subseteq (D \times S)$, and R a single topological variation, $R \in (M \times S)$.

Definition 5 (Triggering pattern support) We say that $Q_1 = \langle X_1, \ldots, X_k\rangle$ is a sub-sequence of $Q_2 = \langle Y_1, \ldots, Y_\ell\rangle$, $Q_1 \preceq Q_2$, if there exists $\langle i_1, \ldots, i_k\rangle$ a strictly increasing sequence in $\mathbb{N}$ such that $X_j \subseteq Y_{i_j}, \forall j = 1, \ldots, k$. The support of a pattern $P = \langle L, R\rangle$ is the set of vertices for which P is a sub-sequence of their descriptive sequence: $\text{SUPP}(P, \Delta) = \{v \in V \mid P \preceq \delta(v)\}$.

Example 4 With $P_1 = \langle\{a^+\}, \{deg^+\}\rangle$ and $P_2 = \langle\{a^+, b^+\}, \{c^-\}, \{deg^+\}\rangle$, it follows that $\text{SUPP}(P_1, \Delta) = \{u_1, u_3\}$ and $\text{SUPP}(P_2, \Delta) = \{u_1, u_3\}$.

We adapt a well known measure, the growth-rate [1] [9,24], to characterize patterns $\langle L, R\rangle$ whose left part L strongly discriminates the topological variation R.

Definition 6 (Triggering pattern growth rate) Let $P = \langle L, R\rangle$, we denote by $\Delta^R \subseteq \Delta$ the set of vertex descriptive sequences that contain R. The growth rate of P is given by:

$$\text{GR}(P, \Delta^R) = \frac{|\text{SUPP}(L, \Delta^R)|}{|\Delta^R|} \times \frac{|\Delta \setminus \Delta^R|}{|\text{SUPP}(L, \Delta \setminus \Delta^R)|}$$

The triggering pattern mining problem. Given a dynamic attributed graph $\mathcal{G}$, a minimum growth rate threshold $minGR$ and a minimum support threshold $minSup$, the problem is to find all triggering patterns $P = \langle L, R\rangle$ such that $|\text{SUPP}(P, \Delta)| \geq minSup$ and $\text{GR}(P, \Delta^R) \geq minGR$. The support measure is defined by the vertices that satisfy the pattern, while the growth rate gives the discriminating power of the sequence variations L to explain a topological change R.

4 Non redundant triggering patterns with semantics

The triggering pattern problem is actually defined in the previous section as a well known frequent sequential pattern mining problem [32]. After a data transformation into a database of (vertex descriptive) sequences, the goal is to find all frequent patterns with the particularity that they must finish with a particular item (a topological change) and being provided with a growth rate higher than a given threshold. However, this comes with several problems and limitations that we develop now and propose solutions to.

4.1 Mining covering triggering patterns

The first main problem is that the support and the growth rate are not enough for producing intelligible patterns: additional measures are required. We propose in this section several examples of meaningful measures. The most important one is the coverage of a pattern (vertex cover [6]): given a graph, the coverage counts the proportion of nodes that can be directly reached from the nodes of the support. The coverage can be generalized to the nodes reachable at a distance n from nodes of the support. When $n = 0$, the coverage is exactly the support. The coverage provides better insights than the support, since it allows to express how the nodes of the support are rooted into a global behavior. However, triggering patterns appear at different time stamps in their supporting descriptive sequences and the set of connected vertices cannot be defined in the dynamic graph: the coverage of a pattern has to be defined up to an aggregated static graph $\mathcal{G}_{aggr}$ that sums-up the connectivity of each vertex along time. In the following we use $\mathcal{G}_{aggr} = (V, \bigcup_{i=1}^{t} E_i)$.

[1] The term growth-rate may be misleading: it is not related to time, but to the appearance of a target attribute in a sub group of objects w.r.t. the rest of the database.

Definition 7 (Coverage of a triggering pattern) Let $\mathcal{G}_{aggr} = (V, E_{aggr})$ be an aggregated graph of the dynamic graph $\mathcal{G}$. The coverage of a pattern P is defined by: $\text{COV}(P, \Delta, \mathcal{G}_{aggr}) = \text{SUPP}(P, \Delta) \cup \{v \in V \mid \exists w \in \text{SUPP}(P, \Delta) s.t. (w, v) \in E_{aggr}\}$.

Example 5 For $P = \langle \{a^+, b^+\}, \{c^-\}, \{deg^+\} \rangle$, we have $cov(P, \Delta, \mathcal{G}_{aggr}) = V$.

The coverage gathers the vertices that have a geodesic distance of 1 with at least one vertex of the pattern support. This measure can be generalized to any geodesic distance value n as follows:

Definition 8 (n-coverage of a triggering pattern) The n-coverage of a pattern P is recursively defined as $\text{COV}^n(P, \Delta, \mathcal{G}_{aggr})$ with $\text{COV}^0(P, \Delta, \mathcal{G}_{aggr}) = \text{SUPP}(P, \Delta)$ and $\text{COV}^i(P, \Delta, \mathcal{G}_{aggr}) = \text{COV}^{i-1}(P, \Delta, \mathcal{G}_{aggr}) \cup \{v \in V \mid \exists w \in \text{COV}^{i-1}(P, \Delta, \mathcal{G}_{aggr}) \; s.\,t. \; (w, v) \in E_{aggr}\}$.

The n-coverage measure is important for two reasons: (1) It conveys *per se* insights on the possibilities for dissemination of a pattern; (2) It makes it possible the definition of additional interestingness measures to focus on specific kinds of triggering patterns. Since the coverage is a generalization of the support,we propose to mine *covering triggering patterns*, i.e. patterns whose coverage is higher than a given threshold $minCov$ and provided with a growth rate higher than $minGR$.

4.2 Additional measures that convey semantics and constraints

The definition of coverage enables to define several measures and constraints on the pattern domain. This is interesting for two reasons: firstly, triggering patterns are provided with more semantics, secondly, the expert can specify some constraints on the patterns to drive her analysis and partly face the classical pattern flooding problem. Indeed, the n-coverage of a triggering pattern can be viewed as a set of nodes inducing a subgraph of $\mathcal{G}_{aggr}$. Constraining such subgraphs makes it possible to mine topological changes whose coverage has a specific structure. We can distinguish two types of constraints, those that rely on the vertices (see Definition 9) and the others that restrict the structure of the subgraph (see Definition 10).

Definition 9 (Constraint on vertices) Let μ be a centrality measure of a vertex v in the aggregated graph $\mathcal{G}_{aggr}$, such as:

- The vertex degree: $deg(v, \mathcal{G}_{aggr}) = |\{u \in V, \; \{u, v\} \in E_{aggr}\}|$
- Clustering coefficient: $\text{CLUST}(v, \mathcal{G}_{aggr}) = \frac{2|\{\{u,w\} \in E_{aggr}, \{u,v\} \in E_{aggr} \wedge \{v,w\} \in E_{aggr}\}|}{deg(v)(deg(v)-1)}$
- Closeness centrality: $\text{CLOSE}(v, \mathcal{G}_{aggr}) = \frac{n}{\sum_{u \in V} |shortest\_path_{\mathcal{G}_{aggr}}(u,v)|}$
- Betweeness centrality: $\text{BETW}(v, \mathcal{G}_{aggr}) = \sum_{u,w} \delta_{v \in shortest\_path_{\mathcal{G}_{aggr}}(u,w)}$

The constraint on the vertices of the n-coverage of a triggering pattern is as follows: $\mathcal{C}(P, \Delta, \mathcal{G}_{aggr}, \mu, m, n)$ iff $\forall v \in \text{COV}^n(P, \Delta) \mid \mu(v, \mathcal{G}_{aggr})\} \; \theta \; m$ with $\theta \in \{\leq, <, \geq, >\}$

Definition 10 (Constraint on the subgraph structure) Let ρ be a measure on a subgraph $G(V)$ induced by the set of vertices V in the aggregated graph $\mathcal{G}_{aggr}$. Examples of such measure are:

- Density: $density(V, \mathcal{G}_{aggr}) = \frac{|V \times V \cap E_{aggr}|}{|V| \times (|V|-1)}$
- Diameter: let $d_{G(V)}(u, v)$ be the shortest path length between the vertices u and v in $G(V)$. The diameter of $G(V)$ is thus defined by

$$diam(V, G_{aggr}) \equiv \max_{u,v \in V} d_{G(V)}(u, v)$$

The constraint on the structure of the n-coverage of a triggering pattern is as follows: $\mathcal{C}(P, \Delta, \mathcal{G}_{aggr}, \rho, m, n)$ iff $\rho(\text{COV}^n(P, \Delta), \mathcal{G}_{aggr})\ \theta\ m$ with $\theta \in \{\leq, <, \geq, >\}$

Finally, remember that a pattern may appear at different time stamps in the vertex descriptive sequences that make its support. This is the case in the example of Figure 1: the pattern $\langle\{a^+, b^+\}, \{c^-\}, \{deg^+\}\rangle$ is supported by both vertices u_1 and u_3. However, these appearances are not synchronized. We propose a measure that allows to quantify the notion of synchronicity.

Definition 11 (Synchronicity) Let $P = \langle L, R\rangle$ be a triggering pattern supported by the set of vertices $SUPP(P) = \{u_1, ..., u_n\}$. We introduce the set $A = \{a_1, a_2, ..., a_n\}$ where $a_i \in T$ is the first time stamp where P occurs in the vertex descriptive sequence $\delta(u_i)$. Similarly, we introduce $B = \{b_1, b_2, ..., b_n\} \subseteq T$ where b_i is the index of R in the first occurrence of P in $\delta(u_i)$. The synchronicity measure is given by:

$$sync(P) = \frac{avg(\{|a_i - a_j|\}_{i,j \in 1...n, i \neq j}) + avg(\{|b_i - b_j|\}_{i,j \in 1...n, i \neq j})}{avg(\{|(b_i - a_i) - (b_j - a_j)|\}_{i,j \in 1...n, i \neq j})}$$

where $avg(.)$ is the mean function. The numerator is 0 if the time stamps of the supporting sequences are fully synchronous. The denominator normalizes w.r.t. the average lengths.

4.3 Limiting redundancy with closed triggering patterns

To limit more strongly the problem of pattern flooding, we propose here to define a condensed representation of triggering patterns, to avoid to output them all, while not loosing the information they convey. To this end, we adapt the notion of closed sequential patterns [32] to the case of triggering patterns. This solution is not straightforward and requires to develop a new algorithm called `TRIGAT` in Section 5.

Classical closed sequential patterns are indeed chosen to limit redundancy: any sequential pattern is closed w.r.t. the support if it does not exist a super-sequence with the same support [32]. Furthermore, the growth-rate is always maximized by those closed patterns [25] and most of the constraints given above are necessary to be evaluated on closed patterns only (the case of the anti-monotone constraints, defined hereafter). We show how to define closed triggering patterns w.r.t. to the coverage measure that maximize the GR measure. As triggering patterns are particular sequences ending on a topological variation, one cannot trivially apply any closed sequential pattern mining algorithm. Instead, we prove several properties that allow to define closed triggering patterns as prefix-closed patterns, and that the latter can be defined from closed sequential patterns. Interestingly, the coverage measure still follows an anti-monotone behavior.

Definition 12 (Prefix-closed pattern) We say that a pattern $P = \langle L_P, R\rangle$ is prefix-closed w.r.t. SUPP iff $\nexists Q = \langle L_Q, R\rangle$ such that $L_P \preceq L_Q$ and $\text{SUPP}(P) = \text{SUPP}(Q)$: P cannot be extended on its left part L_P without changing its support.

In Example 4, P_2 is a closed triggering pattern, i.e. a prefix-closed sequential pattern, while P_1 is not.

Property 1 Let $P = \langle L_P, R\rangle$ and $Q = \langle L_Q, R\rangle$ be two triggering patterns. If $L_P \preceq L_Q$ and $\text{SUPP}(L_P, \Delta^R) = \text{SUPP}(L_Q, \Delta^R)$, then $\text{GR}(L_P, \Delta^R) \leq \text{GR}(L_Q, \Delta^R)$.

Proof We have $\text{SUPP}(L_P, \Delta^R) = \text{SUPP}(L_Q, \Delta^R)$. As $L_P \preceq L_Q$, we obtain that $\text{SUPP}(L_P, \Delta) \supseteq \text{SUPP}(L_Q, \Delta)$ thanks to the anti-monotonicity of the support. By definition 6, we conclude that $\text{GR}(L_P, \Delta^R) \leq \text{GR}(L_Q, \Delta^R)$. □

Property 1 makes it possible to only focus on prefix-closed patterns as they maximize the growth-rate. To extract prefix-closed patterns, we exploit the property asserting that prefix-closed patterns can be retrieved from closed patterns.

Property 2 For any prefix-closed pattern P, there exists a closed pattern Q such that $P \preceq Q$ and $\text{SUPP}(P) = \text{SUPP}(Q)$.

Proof Prefix-closed patterns cannot be extended to the left without changing their support. However, they can be extended to the right while preserving their support. Thus, a closed pattern Q can be seen as a simultaneously prefix and suffix-closed: Q is included in a closed pattern. □

Property 3 states that COV is anti-monotone with respect to $\preceq$ (directly from Definition 7). Since by definition we have that for any pattern P, $\text{SUPP}(P) \subseteq \text{COV}(P)$, one can choose to enumerate the closed triggering patterns based either on support or coverage. This is made precise in the next section.

Property 3 The coverage is anti-monotone w.r.t. $\preceq$, i.e., $\forall P \preceq Q, \text{COV}(P, \Delta, \mathcal{G}_{aggr}) \supseteq \text{COV}(Q, \Delta, \mathcal{G}_{aggr})$.

The same property applies for COV^n. Constraints μ and ρ presented in the previous subsection however, can be or not anti-monotone. When this is the case, we can use a conjunction of several constraints as a main constraint. Indeed, a conjunction of anti-monotone constraints is anti-monotone and it can be also used to prune the search space in the algorithm we develop in the next section [22]. When a constraint is not anti-monotone, we check it in a post-processing by removing afterwards the patterns that do not satisfies it.

5 Mining triggering patterns with the algorithm `TRIGAT`

We introduce `TRIGAT` to mine the complete and correct collection of all coverage-based closed triggering patterns $P = \langle L, R\rangle$ such that $|\text{COV}^n(P, \Delta)| \geq minCov$ and $\text{GR}(P, \Delta^R) \geq minGR$. The algorithm takes benefits from the properties given in Section 4.3 to efficiently prune the pattern search space while pushing anti-monotone

constraints among which the coverage constraint. Algorithm 1 presents the main steps of TRIGAT, including the necessary pre- and post- processing steps. It first builds each vertex descriptive sequence of the dynamic graph (line 1). Covering 1-item sequences, that are sequences of a single descriptor variation satisfying the coverage constraint, are computed in one scan on Δ (line 2) and uncovering items are removed for any sequence of Δ. Then, TRIGAT_ENUM is called. It achieves a depth-first search on a given prefix sequence using a pattern-growth approach that works on *projected databases* according to a prefix sequence s, denoted $\Delta_{|s}$ which returns all the suffixes of s in Δ [32]. TRIGAT_ENUM exploits the pruning techniques based on closed patterns (lines 1 to 3) while pushing the coverage constraint (line 4). We use the *early termination by equivalence* pruning technique, first proposed in the CloSpan algorithm [32][2]. The coverage is computed w.r.t. the union of the adjacency lists of vertices whose projected sequences along the prefix sequence s is not empty (i.e., $s \preceq \delta(v)$). Finally, as prefix sequences can grow either by adding a single descriptor variation in the last set of s, or by adding a new set made of this single descriptor variation at the end of the sequence, these two patterns are recursively considered (lines 5 to 10). TRIGAT_ENUM returns all covering sequences of prefix $\langle s \rangle$. Algorithm 1 ends with a post-processing of C to retain only closed patterns (line 7). To avoid expensive tests that aim at comparing each sequence of C with other sequences in C, we adopt the fast subsumption checking algorithm [35] using a hashmap with a sparse key distribution. Closed triggering patterns are then built from C (line 8). From lines 9 to 11, prefix-closed sequences ending by a topological variation are built from closed sequences, and the growth-rate is computed (in negligible time, see section 6).

6 Experiments

We provide an empirical evaluation of our methodology. Experiments were performed on 2.5GHz and 16GB of RAM machines and TRIGAT is written in C++[3]. The topological measures considered are: degree, closeness, betweenness, eigenvector, network constraint, clustering coefficient, PageRank, hub score, and authority score [18][4].

6.1 Dynamic attributed graphs:

We experiment on three real-world datasets of different natures to provide a performance study as well as some qualitative results.

[2] This enables to avoid to consider any prefix sequence s' having an equivalent projected database than a sequence s discovered before, i.e., $\Delta_{|s} = \Delta_{|s'}$. Two cases are possible. Either $s' \prec s$ (backward sub-pattern) or $s \prec s'$ (backward super-pattern). In case of backward sub-pattern, the exploration of s' and its descendants is stopped. In case of backward super-pattern, the descendant of s are transplanted to s' instead of exploring an already scanned projected database.

[3] See materials at: http://liris.cnrs.fr/~mplantev/doku/doku.php?id=trigat

[4] Measures computed with SNAP http://snap.stanford.edu/

Algorithm TRIGAT
Require: $\mathcal{G} = \{(V, E_i, F)\}$, $minGr$, $minCov$, $\mathcal{G}_{aggr}$, n, a set C of constraints
Ensure: $\mathcal{P}$ the set of closed triggering patterns
1: $\Delta \leftarrow \{\delta(v) | v \in V\}$
2: $I \leftarrow$ all covering 1-item sequences (items α s.t. $\text{COV}^n(\alpha) \geq minCov$)
3: Filter Δ with only covering 1-item sequences w.r.t. $minCov$ and verifying the anti-monotone constaints from C
4: **for all** $s \in I$ **do**
5: $C \leftarrow$TRIGAT_enum$(s, \Delta_{|s}, \mathcal{G}_{aggr}, minCov, n, C)$
6: **end for**
7: Eliminate non-closed sequences from C
8: $C \leftarrow$ prefix closed patterns $\langle s, X_k \rangle \in C$, s.t.$X_k \in (M \times S)$
9: **for all** $P = \langle s, X_k \rangle \in C$ **do**
10: Add P to $\mathcal{P}$ if GR$(\langle s, X_k \rangle, \Delta^{X^k}) \geq minGr$ and non anti-monotone constraints from C are verified.
11: **end for**

Procedure TRIGAT_ENUM
Require: $s = \langle S_1, \ldots, S_\ell \rangle, \Delta_{|s}, \mathcal{G}_{aggr}, minCov, n, C$
Ensure: C the set of covering sequences of prefix s
1: **if** not closed_based_prunnable(s) **and** s verifies all the anti-monotone constraints from C **then**
2: insert s in C
3: **end if**
4: Scan $\Delta_{|s}$, find each $\alpha \in (D \times S)$ s.t. $\text{COV}^n(\alpha) \geq minCov$ and s can be extended to $\langle S_1, \ldots, S_{\ell-1}, \{S_\ell \cup \alpha\} \rangle$ or $\langle S_1, \ldots, S_\ell, \alpha \rangle$
5: **for all** valid α **do**
6: $s \leftarrow \langle S_1, \ldots, S_{\ell-1}, \{S_\ell \cup \alpha\} \rangle$
7: TRIGAT_enum$(s, D_{|s}, \mathcal{G}_{aggr}, minCov, n, C)$
8: $s \leftarrow \langle S_1, \ldots, S_\ell, \alpha \rangle$
9: TRIGAT_enum$(s, D_{|s}, \mathcal{G}_{aggr}, minCov, n, C)$
10: **end for**

Algorithm 1: The TRIGAT algorithm

DBLP: The Digital Bibliography & Library Project[5] covers an important part of the computer science bibliography. All references published between 1990 and 2010 by 2,723 authors (recording more than 10 publications) are elected among 43 conferences/journals. Two additional attributes sum the publications resp. in *conferences* and *journals*. The dynamic attributed graph, with authors as vertices and edges as co-authoring, entails 9 time stamps of 5 years half-overlapping intervals ([1990-1994],[1992-1996],...,[2006-2010]).

Air traffic: The Research and Innovative Technology Administration (RITA) maintains a public database giving the U.S. air carrier traffic statistics[6]. We derive three dynamic graphs whose vertices are the airports, the vertex attributes describe traffic aspects (number of departures/arrivals, number of cancelled flights, number of flights diverted at destination, the mean delay of departure/arrival and the ground waiting at time departure/arrival) and the edges represent air lines. The 3 resulting dynamic attributed graphs differ by their time scale: in *RITA1*, attribute values are summed over days of September 2001; in *RITA2*, the values are accumulated

[5] http://www.informatik.uni-trier.de/~ley/db/
[6] http://http://transtats.bts.gov/

over months from September 2000 to September 2002; and in *RITA3*, they are aggregated over weeks between 01/08/2005 and 25/09/2005 (Katrina hurricane).

Social bookmarking: Del.icio.us[7] provides social networking, bookmarking, and tagging information [5]. In the resulting dynamic graph, edges represent mutual fan relationships. Users are described by the numbers of bookmarks they shared for different categories (*art*, *car*, *student*, etc.). Due to the long tail principle, only categories used at least by 500 are kept. Timestamps are aggregated by year on a period spanning from 2006 to 2010 for the quantitative experiments.

Before extracting triggering patterns from these dynamic graphs, we need to characterize the strength of the attribute value variations as stated in Definition 2. As we are interested in significant variations of vertex attribute values w.r.t. the proper history of each vertex between consecutive timestamps, we consider, for each attribute $d \in D$ and vertex $v \in V$, the set $\{d(v,i) - d(v,i-1),\ 1 < i \leq t\}$ of time derivatives of $d(v)$. Such a discrete set of values can be characterized by its mean $\overline{d(v)}$, that refers to the central value of the set, and its standard-deviation $std(d(v))$ that gives a hint to the homogeneity or heterogeneity of the set. Based on these values, we devised two discretization functions $discr_1$ and $discr_2$:

$$discr_1(v,d,i) = \begin{cases} ++ \text{ if } d(v,i) - d(v,i-1) \geq \overline{d(v)} + 3 \cdot std(d(v)) \\ + \text{ if } d(v,i) - d(v,i-1) \geq \overline{d(v)} + std(d(v)) \\ - \text{ if } d(v,i) - d(v,i-1) \leq \overline{d(v)} - std(d(v)) \\ -- \text{ if } d(v,i) - d(v,i-1) \leq \overline{d(v)} - 3 \cdot std(d(v)) \\ \emptyset \text{ otherwise} \end{cases}$$

$$discr_2(v,d,i) = \begin{cases} + \text{ if } d(v,i) - d(v,i-1) \geq \overline{d(v)} + 3 \cdot std(d(v)) \\ - \text{ if } d(v,i) - d(v,i-1) \leq \overline{d(v)} - 3 \cdot std(d(v)) \\ \emptyset \text{ otherwise} \end{cases}$$

$discr_2$ is used on **RITA1** and **RITA2** air traffic networks since we aim at characterizing a locally high-impact event. $discr_1$ is used on **DBLP**, **RITA3** and **del.icio.us** dynamic graphs where the temporal variations of values are much more progressive. For the coverage computation, we use in these experiments the aggregated graph $\mathcal{G}_{sum}$.

Table 1 reports the main characteristics of the studied dynamic graphs. $|V|$, $|F|$ and $|T|$ are respectively the number of vertices, attributes and timestamps. $|\mathcal{D}|$ is the number of descriptor variations, $\overline{S}$ is the average number of descriptor variations per vertex, $\overline{I}$ is the average number of timestamps having more than one variation. $\overline{\deg}_{sum}$ (resp. density_{sum}) is the average degree (resp. density) of $\mathcal{G}_{sum}$.

Finally, observe that a topological variation may appear several times in a vertex descriptive sequence, for example a vertex may experience a large increase in its degree twice over time. We assume that each occurrence of a topological variation in $\delta(v)$ is distinct from the others: adding an index to those variations is enough to differentiate them and does neither change the problem definition or its resolution.

[7] `http://www.delicious.com/`

	$\|V\|$	$\|F\|$	$\|T\|$	$\|\mathcal{D}\|$	$\overline{S}$	$\overline{I}$	$\overline{\deg}_{sum}$	density_{sum}
DBLP	**2723**	45	9	360	**34.4**	**6.6**	14.7	0.005
RITA1	220	8	**30**	30	16.3	5.1	15.7	**0.07**
RITA2	234	8	24	39	4.5	1.8	17	**0.07**
RITA3	280	6	8	87	28.3	6.5	**15.9**	0.05
del.icio.us	1867	**121**	5	**400**	31	1.6	11	0.003

Table 1: Main characteristics of the dynamic graphs (for each column, the maximum value is in bold).

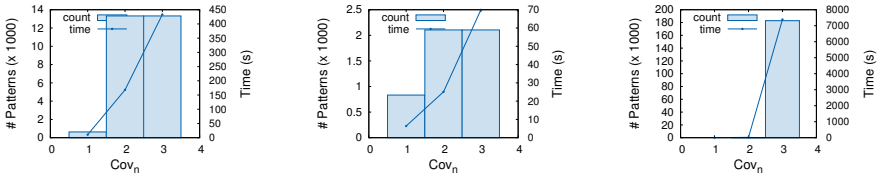
6.2 Quantitative experiments:

In this section, we aim to provide some answers to the following questions regarding scalability and efficiency of TRIGAT:

- What is the impact of the *minCov* parameter on both the running time and the number of triggering patterns?
- What is the impact of the generalized coverage on both the running time and the number of triggering patterns?
- How is the execution time distributed among the variation calculation (pre-processing), the pattern mining step, and the growth rate and the non anti-monotone constraint calculation (post-processing)?
- How are the patterns distributed w.r.t. their support?
- How are the patterns distributed over the 2-dimensional space $(Support, Growth\text{-}rate)$?
- How robust is our approach w.r.t. the number of vertices?

For all quantitative experiments, we set $minGR = 0$ and do not use non anti-monotone constraints to be fair on the reported results, since they are computed in a post-processing.

Impact of the generalized coverage: We first observe on Figure 2 the impact of the generalized coverage on the running time and the number of discovered patterns. Obviously, considering a more general definition of the coverage drastically increases the search space. Therefore, the number of patterns and the time consumption highly increases when considering generalized coverage. Two points are worth to be discussed regarding this experiment. First, setting n to 2 or 3 allows the discovery of patterns that would not have been discovered otherwise, i.e., when $n = 1$. Second, as it can be seen on Figures 2a and 2b, the same patterns are mined whatever the value of n when $n > 1$ but with a much more higher time consumption when n goes up. This point shows that vertices supporting these patterns are member of some very dense parts of the graph, i.e, the diameter of each connected components within the subgraph induced by the supporting vertices equals 2. However we will see in the qualitative interpretation of the results that the integration of more semantic constraints on vertices and on the subgraph structure during the mining phase compensates for this point and limit the number of triggering patterns. Note that in Figure 2 the minimum coverage thresholds $minCov$ are chosen such that the computations are feasible for $n \in \{1, 2, 3\}$.

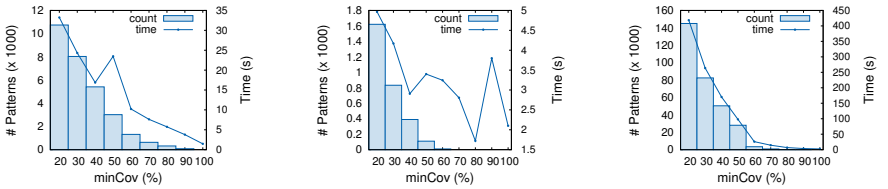


(a) RITA1 (minCov=70%) (b) RITA2 (minCov=30%) (c) RITA3 (minCov=100%)

(d) DBLP (minCov=90%) (e) Del.ico.us (minCov=70%)

Fig. 2: Impact of the generalized coverage.

Impact of *minCov* parameter: As stated in the previous results, considering $n > 1$ without additional constraints can lead to non-tractable experiments. For this reason, we consider $n = 1$ in the rest of the quantitative experiments. We observe on Figure 3 the impact of the $minCov$ value on the running time and the number of discovered patterns. The pruning ability of the $minCov$ parameter is verified. The figures show that considering low $minCov$ values is difficult, but Figure 4 provides the explanation: it often happens that the support is much more smaller than the coverage.



(a) RITA1 (b) RITA2 (c) RITA3

(d) DBLP (e) Del.ico.us

Fig. 3: Impact of $minCov$ parameter (considering Cov^1).

Runtime distribution: TRIGAT consists of three main steps: The discretization (pre-processing step), the discovery of closed triggering patterns (mining step), and the computation of the growth rate values and non anti-monotonic constraint calculation and pruning (post-processing step). Figure 5 reports, for each dynamic graph, the allocation of the running time among these three steps. We observe that the post processing is negligible whatever the considered dynamic graph and $minCov$ value. The lower $minCov$, the more time is spent to mine patterns. Reversely, for high $minCov$ values, the discovery of closed triggering patterns is very fast. Besides, since the time needed to compute the evolutions (discretization) is constant regardless $minCov$, this step is predominant for high $minCov$ values (since the mining step is fast).

Distribution of pattern supports: Figure 6 reports the distribution of the discovered patterns w.r.t. the support. The $minCov$ value has been chosen to be the lowest feasible value for each dynamic graph accordingly to Figure 3. We observe that most of the extracted patterns have a very low support size compared to their coverage one. This point is even more obvious on the DBLP dynamic graph since many patterns have a very low support, i.e., less than 1%. This is particularly interesting since we can mine triggering patterns having a very low support without considering the traditional $minSupp$ constraint. Furthermore, these covering patterns are insightful w.r.t. the growth rate. Indeed, we now aim at studying *how triggering the discovered patterns are* by visualizing the triggering patterns within the 2-dimensional space $(Support, Growth\ rate)$. Intuitively, the higher the growth rate of the pattern $\langle L, R\rangle$, the more discriminative is the L part for the topological variation R. Figure 7 reports our results. Patterns with an infinite growth rate (∞) are called *jumping patterns*, their sequence L only appears in sequences of Δ^R. Such a pattern is of crucial interest since it means that whenever a vertex supports the pattern $\langle L\rangle$, it also supports the triggering pattern $\langle L, R\rangle$. Interestingly, apart from the DBLP dynamic graph, jumping patterns have been discovered in every dynamic graph. In the DBLP dataset however, the growth-rate can reach an important value of 100. Conversely, triggering patterns, that have a low growth-rate, are of less interest and can be removed during the final step (post-processing). TRIGAT can discover patterns simultaneously highly discriminative and covering, either with high or low support.

Impact of the number of vertices on the execution time: We evaluate the robustness of TRIGAT w.r.t. $|V|$. Each of the five dynamic graphs have been replicated

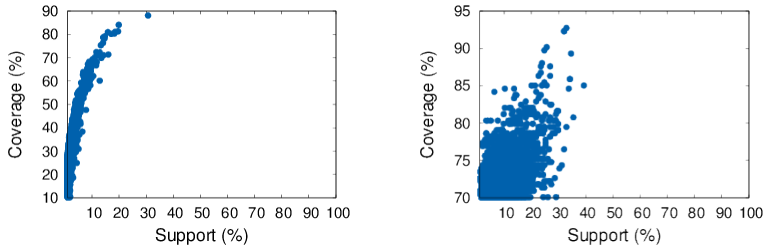
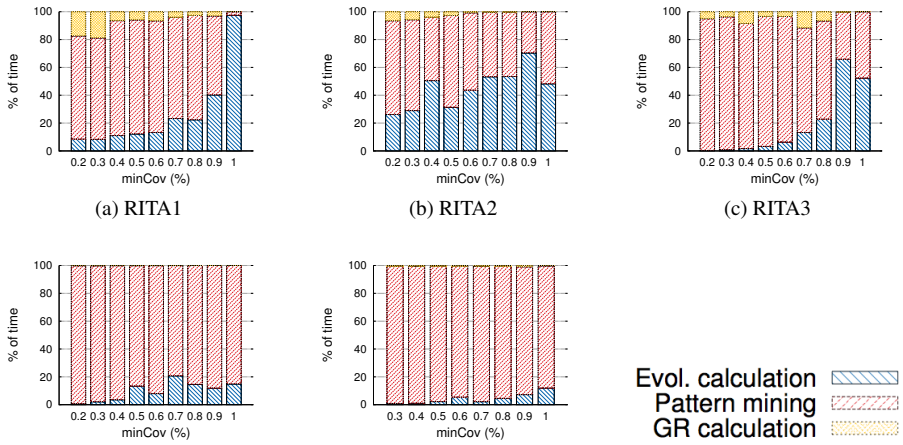


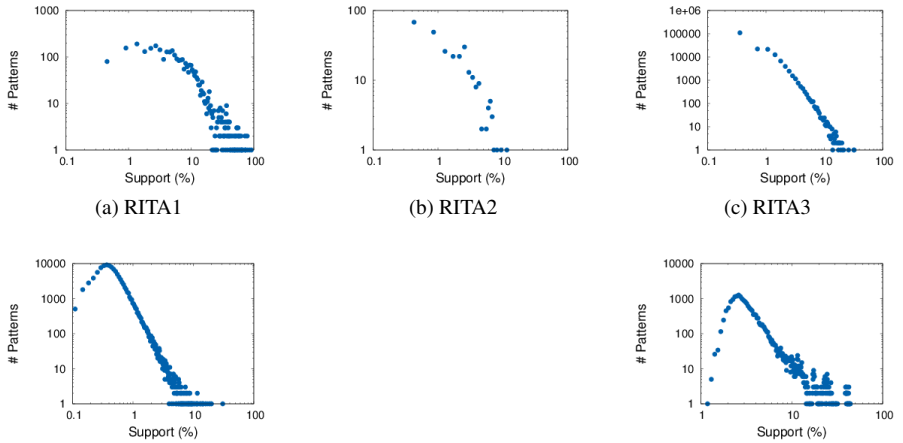
Fig. 4: Distribution of coverage and support for datasets DBLP (minCov=10%, left) and RITA1 (minCov=70%, right).



(a) RITA1 (b) RITA2 (c) RITA3

(d) DBLP (e) Del.ico.us (f) Legend

Fig. 5: Runtime distribution.

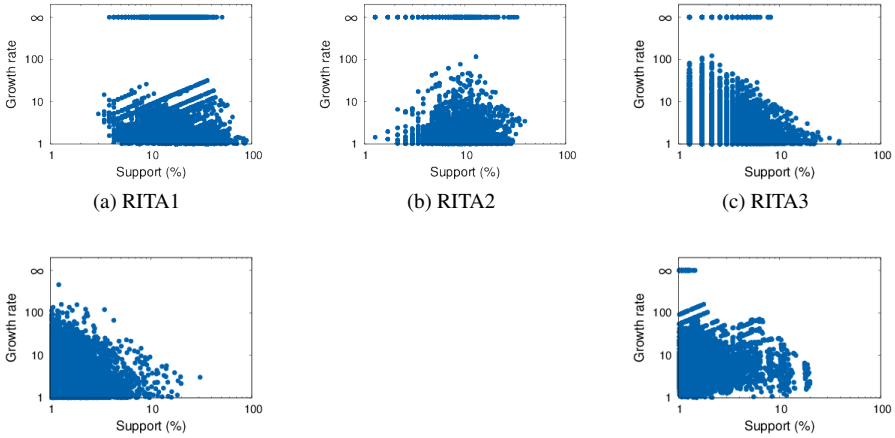


(a) RITA1 (b) RITA2 (c) RITA3

(d) DBLP (e) Del.ico.us

Fig. 6: Distribution of pattern supports.

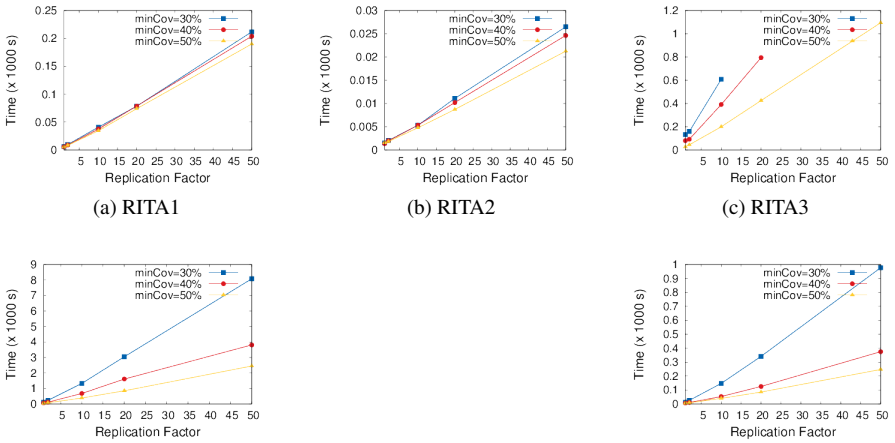
as follows. Let r be a replication factor, each graph is copied r times resulting in r disconnected graphs at each timestamp: it is guaranteed that the set of triggering patterns remains identical whatever the value of r. Figure 8 reports experiment results with different $minCov$ values. The results are very similar for all datasets. As expected, the running time is linearly correlated with the replication factor. The higher the $minCov$, the stronger is the slope of the curve: `TRIGAT` can deal with dynamic graphs having up to 150K vertices.



(a) RITA1 (b) RITA2 (c) RITA3

(d) DBLP (e) Del.ico.us

Fig. 7: Growth rate vs support.



(a) RITA1 (b) RITA2 (c) RITA3

(d) DBLP (e) Del.ico.us

Fig. 8: Scalability test with n dates replication.

6.3 Qualitative interpretation:

We now present some triggering patterns $\langle L, R\rangle$, written $L \rightarrow R$ to ease reading, and study their usefulness on the aforementioned networks.

DBLP: We set the minimal coverage threshold to 10%. One may argue that this threshold is too high when dealing with real-world data. However, we have shown that the support of a pattern can be much more smaller than its coverage: In Figure 9,

Rank	Pattern	Support	Coverage	Growth rate	α
1	$\{closeness_1^-\}, \{IEEETransKnowlDtEn^+\}, \{numCliques_1^+\} \rightarrow \{numCliques_1^-\}$	15	578	87.4	38.5
2	$\{clustering_1^{++}, degree_1^{++}\}, \{Journal^{++}, eigenvector_2^{++}\} \rightarrow \{eigenvector_3^{++}\}$	31	546	71.6	17.6
3	$\{ICDE^+, numCliques_1^+\} \rightarrow \{numCliques_1^-\}$	22	606	64.1	27
4	$\{eigenvector_1^{++}, degree_1^{++}\}, \{VLDB^{++}, degree_2^{++}\} \rightarrow \{degree_3^{++}\}$	29	580	63.8	20
5	$\{eigenvector_1^{++}, clustering_1^{++}\}, \{Journal^{++}, eigenvector_2^{++}\} \rightarrow \{eigenvector_3^{++}\}$	36	619	59.3	17.19
6	$\{ACMTransDBSys^+\}, \{numCliques_1^+\} \rightarrow \{numCliques_1^-\}$	20	547	58.3	27.35
7	$\{eigenvector_1^{++}\}, \{Journal^{++}, betweennes_3^{++}\} \rightarrow \{betweennes_4^{++}\}$	20	587	58.4	29.35
8	$\{eigenvector_1^{++}\}, \{VLDB^{++}, degree_2^{++}\} \rightarrow \{degree_3^{++}\}$	30	623	56.47	20.7
9	$\{SIGMOD^-\}, \{numCliques_1^+\} \rightarrow \{numCliques_1^-\}$	32	754	53.3	23.56
10	$\{closeness_1^-\}, \{SIGMOD^-\}, \{numCliques_1^+\} \rightarrow \{numCliques_1^-\}$	18	552	52.4	30.6

Table 2: Top ten patterns of the DBLP co-authorship network according to the growth rate value.

the $minCov = 10\%$ threshold induces the extraction of patterns with a support less than 1% ($minGR = 1$). The mining task (including pre/post processing) takes 307 seconds and returned $3,261$ patterns. The growth rate distribution for those patterns ranges from 1 to 87, with a mean value of 4.5 and standard deviation of 3.6. Table 2 gives the top-10 patterns with a growth rate greater than 50.

Consider now a measure given by the ratio $\alpha(P, \Delta) = \frac{|\text{COV}(P,\Delta,\mathcal{G}_{aggr})|}{|\text{SUPP}(P,\Delta)|} \in [1, |V|]$, which allows to distinguish the patterns supported by a group of isolated vertices (values close to 1) to the ones supported by very connected vertices (much higher values than 1). The support of each pattern given in Table 2 is in average around 1% and the coverage around 22%, thus a high α value.

These patterns explicit the conferences or journal venues that strongly impact and/or explain one's collaborations or authority in the DBLP network. For example, the first pattern tells us about the impact of publishing more in the IEEE Transactions on Knowledge and Data Engineering. The Figure 9 (left) gives the subgraph induced by the support (labeled in red), and the coverage: the high density reveals a community aspect. In contrast, the graph of pattern 8 is much more sparser (Figure 9 right) which may reflect that publishing at VLDB triggers a positive variation of the author degree in the graph, for authors that cover well the graph (high coverage) but do not collaborate together (low density). As such, one can be interested in querying about a specific conference. Let us continue with the example of the VLDB conference: we wonder if publishing in VLDB, and probably presenting there, could be an interesting start for increasing one's degree, i.e. making collaborations. We query the top-k patterns w.r.t. growth rate that involve VLDB in the left hand-side and the degree in the right-hand side. All the patterns returned have a growth rate higher than 20. Consider now the IEEE ICDM conference: the top pattern that involves it has a growth rate of 5 and tells us that publishing at ICDM may help to be more central in the co-authoring graph. The lower growth rate (5) when comparing with VLDB (20) can be explained by the fact that ICDM is younger than VLDB.

RITA1: With $minCov = 10\%$, `TRIGAT` extracts $1,922$ jumping patterns (i.e. having an infinite growth rate value). To retrieved the most influential airports that were disturbed by $09/11$ event, we compute the ratio α of coverage divided by support: the higher this ratio, the more potentially influential. The top extracted pattern is $\{\#Canceled^+\}\{Degree^-, Closeness^-, NumCliques^-, Pagerank^-, Betweennes^-\} \rightarrow Degree^+$ which has a support of 5, a coverage of 144, and $\alpha = 28.8$, i.e. in average an air-

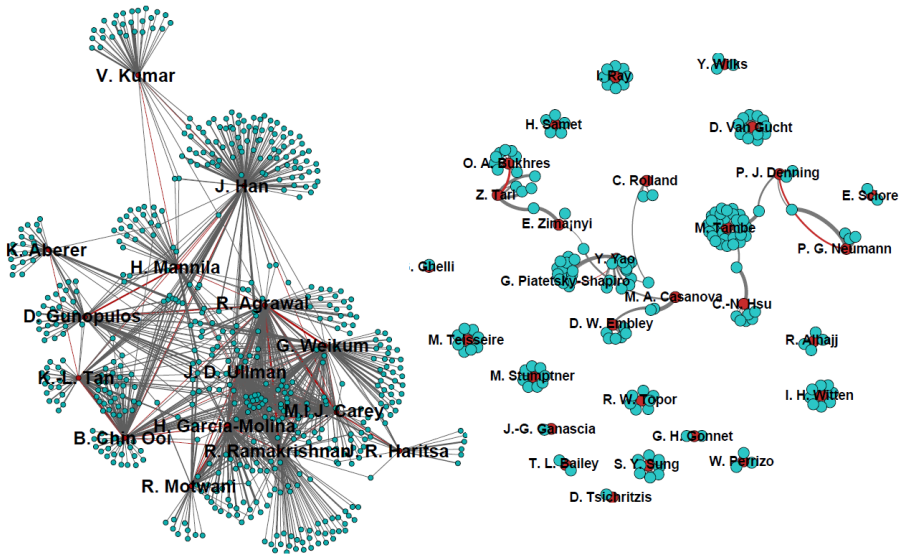


Fig. 9: The patterns ranked 1^{st} (left) and 8^{th} (right) from the DBLP network w.r.t. growth rate. Labeled vertex (in red) represent the support, while all vertices form the coverage.

port of the support is connected to 29 airports that do not belong to the support. By considering the sequences of Δ that support this pattern, we observe that the two first itemsets always appear on days 11 and 12 while the last itemset appears on days 13 or 14. This pattern tells us that between September 11 and 12 an increase of canceled flights has been observed, followed the next day by a decrease of several centrality indexes, which triggered a degree increase one or two day later[8] .

RITA2: This dynamic graph is set up to analyze the impact of 09/11 event on the US air traffic with a higher temporal granularity and a larger interval of time surrounding 09/11. We set the minimum coverage threshold to 10% and extract 278 patterns and keep the 208 jumping ones. As we *a priori* know that 09/11 led to an important number of canceled flights, we only retain the 25 patterns that contain this descriptor variation and sort them by support. For the 20 first patterns, it appears that the increase of canceled flights arises at the time of the tragedy. For example, the pattern $\{\#Canceled^+\}\{\#Canceled^-\}, \{numCliques^-, Betweeness^+\}$ $\rightarrow numCliques^+$ with (support=8, coverage=61) means that 8 airports, covering more than 25% ($\frac{61}{234}$) of the network, underwent an increase of their canceled flights, followed in another month by a decrease of this quantity and next by a topological variation. This lead some months later to an increase of the number of cliques. Now we should pay attention to when those variations happened. The first itemset always happened in September 2001. The second appeared at different months between De-

[8] As reported at `http://en.wikipedia.org/wiki/Closings_and_cancellations_following_the_September_11_attacks#North_American_airspace`

cember 2001 and January 2002 while the last itemset arose the month after the second itemset. In all the cases, the triggered topological variation happened in March 2002, which suggest a back to the normal of the US air traffic, which is not contradicted by the IATA conclusions[9].

RITA3: On this network, TRIGAT extracts $185,486$ patterns with $minCov = 10\%$. To observe the impact of the Katrina hurricane on the US airport traffic, we filter the patterns with a syntactic constraint enforcing an increase on the number of canceled flights and on the number of diverted flights. It selects $7,427$ patterns among which 16 have a support higher than 5. Their coverage varies between 25% and 35%, with small growth rate, except the jumping pattern: $\{Cancelled^+, DelayAtDeparture^+\}$, $\{\#Diverted^-, \#Departure^-, \#Arrival^-\} \rightarrow \{closeness_1^-\}$. All the airports supporting this pattern are located in the US West coast where Katrina raged. The itemset $\{\#Diverted^-, \#Departure^-, \#Arrival^-\}$ happened during the week of Katrina's highest activity.

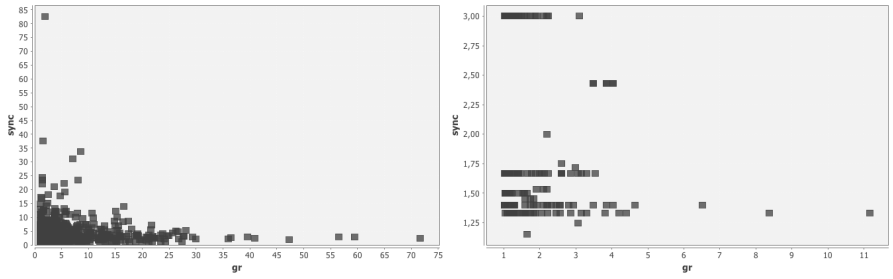
del.icio.us: We are here interested to discover topics of bookmarks (i.e. user tags) that may trigger a topological change. For example, if a user has more bookmarks of a given topic, does it make this user more followed by the others? We set the minimum coverage to 10% and extract $27,048$ patterns among which $1,424$ ones conclude on a positive variation of a topological attribute. We ranked them w.r.t their growth rate. The 30 first patterns have a growth rate higher than 2.5 with a maximum of 12 and an average of 4.16. Support and coverage cardinalities are roughly the same with an average of respectively 13 and 198 (out of 1867 users). Each pattern contains at most two attributes that denotes a variation of a topic, and there is no negative variation. Interestingly, most of those topics are either among *how-to*, *tutorial*, *web design*, *visualization*, which denote probably a "teaching triggering", or among *video* and *community* which are more "social triggering".

Synchronized and non synchronized patterns. TRIGAT is able to identify *synchronous* triggering patterns, whose vertex attributes change at the same timestamps in the supporting sequences of Δ. This phenomenon was observed in RITA networks. Besides, TRIGAT is also able to discover *asynchronous* triggering patterns, whose vertex attributes change at different timestamps in the supporting sequences. This is the case for the DBLP network, since it depicts scientific careers of researchers with different experiences (PhD student, junior scientist, professor, etc.). This can be observed on Figure 10: the RITA1 patterns (with a growth rate higher than 1) have a synchronicity between 1 and 3 while it is more spread for DBLP.

7 Conclusion

Triggering patterns are sequences of variations of vertex attribute values that may *trigger* topological changes in a dynamic attributed graph. Our methodology relies on closed sequence mining, where each vertex is encoded by a sequence of its variations. To assess the interest of a pattern, both growth-rate and coverage are studied:

[9] http://www.iata.org/pressroom/documents/impact-9-11-aviation.pdf



(a) DBLP (b) RITA1

Fig. 10: Synchonicity measure (sync Y-axis) vs. grow rate (gr X-axis) for patterns extracted from the DBLP dataset with $minCov = 20\%$ (left) and the RITA1 dataset with $minCov = 20\%$ (right).

the growth-rate gives the discrimination power of a sequence of variations to explain a topological change, while the coverage tells us about the diffusion potential of the attributes changes towards vertices of its neighborhood. Several measures on the patterns (and associated constraints) provide the pattern with semantics and are useful in application scenarios. We experimented with this original approach, and demonstrate the capability of triggering patterns to explain topological changes by attribute variations in real-world dynamic graphs. These case studies show the capability of `TRIGAT` to discover sensible patterns in feasible time.

References

1. R. Ahmed and G. Karypis. Algorithms for Mining the Evolution of Conserved Relational States in Dynamic Networks. In *ICDM*, pages 1–10. IEEE, 2011.
2. M. Berlingerio, F. Bonchi, B. Bringmann, and A. Gionis. Mining Graph Evolution Rules. In *ECML/PKDD*, pages 115–130, 2009.
3. K. M. Borgwardt, H.-P. Kriegel, and P. Wackersreuther. Pattern mining in frequent dynamic subgraphs. In *ICDM*, pages 818–822. IEEE, 2006.
4. B. Bringmann, M. Berlingerio, F. Bonchi, and A. Gionis. Learning and predicting the evolution of social networks. *IEEE Intelligent Systems*, 25(4):26–35, 2010.
5. I. Cantador, P. Brusilovsky, and T. Kuflik. Information heterogeneity and fusion in recommender systems. In *RecSys*. ACM, 2011.
6. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (3. ed.)*. MIT Press, 2009.
7. P. O. V. de Melo, C. Faloutsos, and A. A. F. Loureiro. Human dynamics in large communication networks. In *SDM*, pages 968–879. SIAM, 2011.
8. E. Desmier, M. Plantevit, C. Robardet, and J.-F. Boulicaut. Trend mining in dynamic attributed graphs. In *ECML/PKDD*, pages 654–669, 2013.
9. G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *KDD*, pages 43–52, 1999.
10. L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.
11. M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.
12. A. Goyal, F. Bonchi, L. V. S. Lakshmanan, and S. Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social Netw. Analys. Mining*, 3(2):179–192, 2013.

13. S. Günnemann et al. Subspace clustering meets dense subgraph mining: A synthesis of two paradigms. In *ICDM*, pages 845–850, 2010.
14. A. Inokuchi and T. Washio. Mining frequent graph sequence patterns induced by vertices. In *SDM*, pages 466–477. SIAM, 2010.
15. M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, and S. Yang. Social contextual recommendation. In *CIKM*, pages 45–54, 2012.
16. A. Khan, X. Yan, and K.-L. Wu. Towards proximity pattern mining in large graphs. In *SIGMOD*, pages 867–878. ACM, 2010.
17. M. Lahiri and T. Y. Berger-Wolf. Mining periodic behavior in dynamic social networks. In *ICDM*, pages 373–382. IEEE, 2008.
18. J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *WWW*, pages 695–704. ACM, 2008.
19. J. Leskovec and R. Sosič. SNAP: A general purpose network analysis and graph mining library in C++. `http://snap.stanford.edu/snap`, June 2014.
20. F. Moser, R. Colak, A. Rafiey, and M. Ester. Mining cohesive patterns from graphs with feature vectors. In *SDM*, pages 593–604. SIAM, 2009.
21. P.-N. Mougel, C. Rigotti, M. Plantevit, and O. Gandrillon. Finding maximal homogeneous clique sets. *Knowledge and Information Systems*, pages 1–30, 2013.
22. R. T. Ng, L. V. S. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained association rules. In L. M. Haas and A. Tiwary, editors, *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA.*, pages 13–24. ACM Press, 1998.
23. K.-N. Nguyen, L. Cerf, M. Plantevit, and J.-F. Boulicaut. Discovering descriptive rules in relational dynamic graphs. *Intell. Data Anal.*, 17(1):49–69, 2013.
24. P. K. Novak, N. Lavrač, and G. I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *J. Mach. Learn. Res.*, 10:377–403, June 2009.
25. M. Plantevit and B. Crémilleux. Condensed representation of sequential patterns according to frequency-based measures. In *Adv. in Intelligent Data Analysis*, LNCS (5772), pages 155–166. Springer, 2009.
26. A. Prado, B. Jeudy, É. Fromont, and F. Diot. Mining spatiotemporal patterns in dynamic plane graphs. *Intell. Data Anal.*, 17(1):71–92, 2013.
27. A. Prado, M. Plantevit, C. Robardet, and J.-F. Boulicaut. Mining graph topological patterns: Finding co-variations among vertex descriptors. *IEEE TKDE*, 99:1, 2012.
28. C. Robardet. Constraint-based pattern mining in dynamic graphs. In *ICDM*, pages 950–955. IEEE, 2009.
29. J. Sese, M. Seki, and M. Fukuzaki. Mining networks with shared items. In *CIKM*, pages 1681–1684. ACM, 2010.
30. A. Silva, W. Meira, and M. J. Zaki. Mining attribute-structure correlated patterns in large attributed graphs. *PVLDB*, 5(5):466–477, 2012.
31. H. Tong, S. Papadimitriou, J. Sun, P. S. Yu, and C. Faloutsos. Colibri: fast mining of large static and dynamic graphs. In *KDD*, 2008.
32. X. Yan, J. Han, and R. Afshar. Clospan: Mining closed sequential patterns in large databases. In *SDM*, pages 166–177. SIAM, 2003.
33. Y. Yang, J. Yu, H. Gao, J. Pei, and J. Li. Mining most frequently changing component in evolving graphs. *WWW*, pages 1–26, 2013.
34. C. H. You, L. B. Holder, and D. J. Cook. Learning patterns in the dynamics of biological networks. In *KDD*, pages 977–986, 2009.
35. M. J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed itemset mining. In *SDM*. SIAM, 2002.
36. Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki. Attributed graph mining and matching: An attempt to define and extract soft attributed patterns. In *CVPR*, pages 1394–1401, 2014.
37. Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment*, 2(1):718–729, 2009.